

CAAP Quarterly Report

Date of Report: *April, 7th 2020*

Contract Number: *693JK31850007CAAP*

Prepared for: *Robert Smith, Project Manager, PHMSA/DOT*

Project Title: *A novel structured light based sensing and probabilistic diagnostic technique for pipe internal corrosion detection and localization*

Prepared by: *Mohand Alzuhiri, Rahul Rathnakumar, Dr. Yiming Deng, Dr. Yongming Liu*

Contact Information: *Dr. Yiming Deng (MSU) and Dr. Yongming Liu (ASU)*

For quarterly period ending: *April, 7th 2020*

Business and Activity Section

(a)Generated Commitments

Project abstract: Internal corrosion in pipes is dangerous due to multiple factors contributing to its development. Degradation of pipeline health is susceptible to hazard due to failure. To prevent such failures, a major challenge for the maintenance crew to detect and repair corrosion still prevails due to difficult and expensive accessibility during scheduled maintenance. The proposed method will focus on the development of novel structural light-based imaging for internal corrosion detection, which simplifies the detection process while achieving superior spatial resolution. The proposed approach will develop an endoscopic structured light scanning tool that is based on phase measurement profilometry (PMP). The developed system will be simple to fabricate and easy to be used by maintenance personnel with minimal skillset due to its intuitive scans. The structured light system will be developed to generate high-resolution reconstructed images representing surface texture with high accuracy. Based on the images, additional processing capabilities developed using Bayesian updating technique will give the capability of automatic classification and identification of different types of precursors. A convolutional neural network-based corrosion detection method will provide automated detection, which further minimizes the operator involvement. The uncertainty quantification technique will be integrated to enhance the probability of detection and to quantitatively determine the damage size and location.

Based on the identified challenges and state-of-the-art techniques, a complete solution is needed to detect, localize and evaluate internal corrosion in metal pipelines. Our proposed solution is to develop a phase measurement profilometry (PMP) structured light-based tool to detect and locate any surface defects and damages on the pipe wall, which includes specifically corrosion. Internal corrosion will be effectively

detected and evaluated by checking the amount of material loss, color change, spread and pattern in the pipe wall simultaneously with a capability of integrating with ILI platforms. This optical inspection is also integrated with a set of numerical tools to evaluate structured light sensor information in order to automate the analysis of inspection data. The specific technical objectives/goals of the proposed research are:

- Design and develop a PMP structured light-based in-line inspection endoscopic scanner. The deliverables include:
 - design a new SL module to produce patterns with high resolution and contrast.
 - develop a new scheme to calibrate the new PMP based projector and camera(s).
- Develop a new reconstruction algorithm called moving phase measurement profilometry (MPMP) to exploit the system movement of the scanner along the pipe to enhance the quality of the reconstruction and detection.
- Evaluate the suitability of different optical methods like stereo cameras to enhance the performance of corrosion detection.
- Develop a convolutional neural network-based model for the automatic detection and classification of corrosion damages from the provided structured light sensor data to mitigate the need for manual analysis of 3D sensor massive data.
- Develop an uncertainty quantification technique to enhance the probability of detection and to quantitatively determine the damage size and location.

Educational Objectives: Another major objective of the proposed effort is to inspire, educate and train Ph.D. and MS students to address pipeline safety challenges, potentially as a career after their graduation. If funded, two Ph.D. students from both universities and several MS/undergraduate students will be included in this CAAP program. They will be trained and educated in science and engineering to address pipeline safety and integrity challenges. The PIs believe education is a critical component of the CAAP project, and we will integrate research with educational activities to prepare the next generation scientists and engineers for the gas and pipeline industry. Specific educational objectives include:

- Inspiring, educating and training the graduate students at MSU and ASU as research assistants for pipe integrity assessment and management. Our previous successful CAAP projects have produced several engineers, researchers and summer intern in gas and pipeline industry,
- Integrating research topics from this effort with the existing undergraduate research programs at MSU, e.g. ENSURE program at the College of Engineering and ASU to involve undergraduate students in pipe safety research.
- Improving the curriculum at MSU (e.g., Nondestructive Evaluation) and ASU (e.g., Machine Learning and Artificial Intelligence) using the scientific findings and achievement from the proposed research,
- Adapt research topics from this project to student projects in seminar, senior design, and project courses, in order to make educational impacts on broader groups of students,

- Encourage the graduate research assistants involved in this project and students in the courses to apply for internships at USDOT/PHMSA and industry to practice their learned skills and gain practical experiences in areas related to pipe safety and integrity.

The above-mentioned goals and objectives of this CAAP project will be well addressed and supported by the proposed research tasks. Development, demonstrations and potential standardization to ensure the integrity of pipeline facilities will be carried out with the collaborative effort among two different universities and our industry partner, Gas Technology Institution. This MSU-ASU-GTI team has successfully completed several PHMSA projects including “Slow Crack Growth” study, which was ranked No. 2 overall in all core PHMSA projects in 2017. The quality of the research results will be overseen by the PIs and DOT program manager and submitted to high-profile and peer-reviewed journals and leading conferences. The proposed collaborative work provides an excellent environment for the integration of research and education as well as tremendous opportunities for two universities supported by this DOT CAAP funding mechanism. The graduate students supported by this CAAP research will be heavily exposed to ILI, NDE, reliability and engineering design topics for emerging pipeline R&D technologies. The PIs have been actively encouraging students to participate in past and ongoing DOT projects and presented papers at national and international conferences. Students who are not directly participating in the CAAP project will also benefit from the research findings through the undergraduate and graduate courses taught by the PIs and attending university-wide research symposium and workshop.

(b) Status Update of Past Quarter Activities

1 Task 1 – Structured Light System Development

This project aims to provide a fast, robust and easy to use tool to detect and characterize corrosion and corrosion-related damage. In this task, we are aiming to fabricate and miniaturize a sensor that can be inserted inside gas pipelines and building a simple interface to guarantee its ease of use. The following sections will show the progress that MSU made in creating a new sensor that can be inserted inside a 6-inch pipe and also shows some initial images from the new setup.

In the past quarters, MSU team worked on developing and testing the reconstruction algorithm by using both simulations and experiments. We also finished developing the sensor hardware for rectangular geometry and completed the designs to adapt the sensor to cylindrical geometry. The proposed MPMP system extracts the shape of the scanned surface by monitoring the phase changes during the system movement. The extraction of the exact phase values requires precise knowledge of the movement of the sensor. Another issue is that a special procedure is needed to be developed to calibrate the projector with the sensor camera. Projector calibration is a relatively difficult process because the projector cannot see it surrounding, therefore a camera is needed to calibrate the projector[1] [2]. Therefore, we opted to include the stereo sensor in the reconstruction process in the last quarter. During the last quarter, we worked on laying the initial framework for the sensor design and the reconstruction process. We also demonstrated the possibility of using the traditional four-step phase-shifting with stereo by using simulations. In this quarter, we worked on achieving the following:

- Demonstrate the feasibility of using the proposed algorithm the system for our MPMP system in a simulation environment.

- Improve the reconstruction process to achieve subpixel accuracy
- Demonstrate the feasibility of the new system design experimentally.
- Collect data for ASU to accelerate the development of their machine learning algorithm

1.1 Algorithm development:

In our proposed framework, the stereo cameras are used to acquire an initial rough estimation of the imaged surface and then the phase is used to refine the initial 3D structure of the scanned object. In this framework, we assume a unique wrapped phase within 2π . This framework eliminates the need for projector calibration, linear phase projection and uniform scanned surface color. A simulation platform was created to validate the algorithm as shown in **Figure 1**. The setup consists of two stereo cameras and a projector that projects a sinusoidal pattern on a specially designed 3D object. CAD software was used to create a flat surface with material loss defects to serve as a testing bench for the algorithm. The defects have cylindrical shapes with diameters that are decreasing with the increase of the depth.

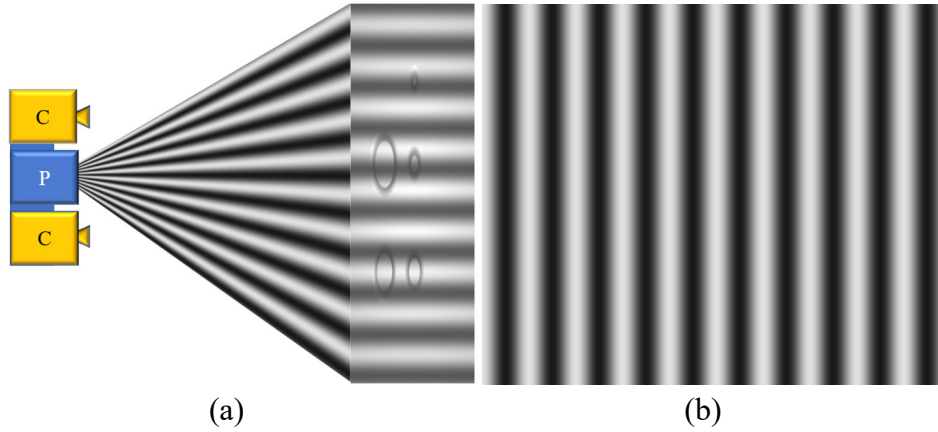


Figure 1: a) Schematic of the simulation setup of the stereo assisted phase-shifting system, b) Sinusoidal pattern with linearly increasing phase distribution

The structured light system is moved in front of the scanned object and four frames are acquired at distances that represent multiple of the $\lambda/4$. Where λ represents the wavelength of the projected sinusoid on the reference plane. Image sequences from the left and right cameras are shown in **Figure 2** and **Figure 3**.



Figure 2: Images sequence from the left camera

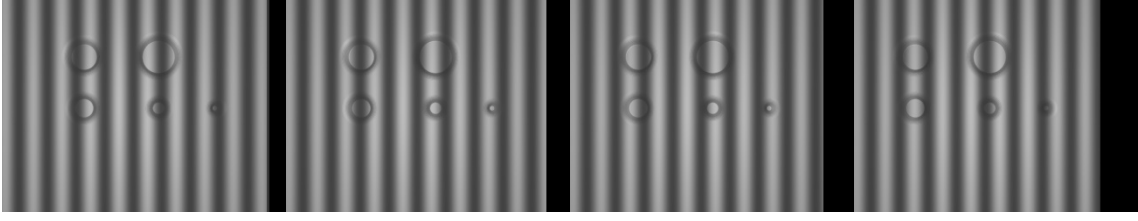


Figure 3: Images sequence from the right camera

The acquired images are registered by cropping multiple of twenty six pixels from each image (assuming a constant speed of 26 pixels for the reference plane). For a phase-shifting system with N projected patterns ($N=4$ in our case), the intensity of each pixel acquired by the camera is given by:

$$I_n = I' + I'' \cos\left(\phi + \frac{2n\pi}{N}\right), \quad n = 0, 1, 2, \dots, N-1,$$

I_n is the intensity of the camera pixel for the n th shifted fringe, I' is the ambient light intensity and I'' represents the modulation signal intensity. A clear 2D image (DC value) without the effect of the sinusoids can be acquired summing the four registered images

$$I_{DC} = \frac{I_1 + I_2 + I_3 + I_4}{4}$$

The DC values of the acquired left and right images are shown in **Figure 4a** and **b**. The DC images show that the distortion from the projected sinusoids was removed to provide a clearer image for the scanned object. These cleaned images serve as an input for stereo comparison where stereo disparity map are acquired by using a block matching algorithm [3]. The initial disparity map from the block matching is shown in **Figure 4c**. The algorithm was able to match the edges of the artificial grooves due to their unique features. The flat surfaces were filled by using the average value of the matched points.

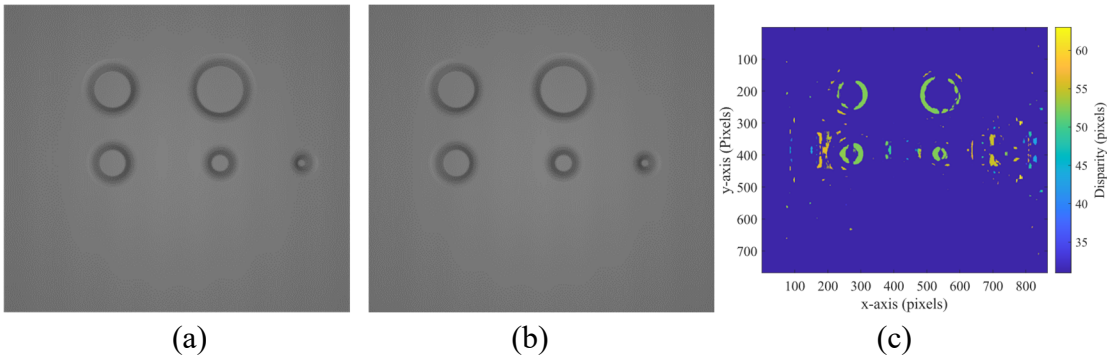


Figure 4: a) Averaging results for the left camera, b) Averaging results for the right camera, c) Disparity map with block matching

The initial disparity map serves a seed point to refine the final disparity map. The phase from each camera can be extracted by solving the set of four linear equations. By assuming a constant ambient light during the scanning, the phase (ϕ) at each image point is given by:

$$\phi(x, y) = \text{atan}^{-1}\left(\frac{I_2 - I_4}{I_1 - I_3}\right),$$

where the arctan function produces a wrapped phase that ranges from $-\pi$ to $+\pi$. By using the geometric constraints, we assume that the phase is unique within the search window.

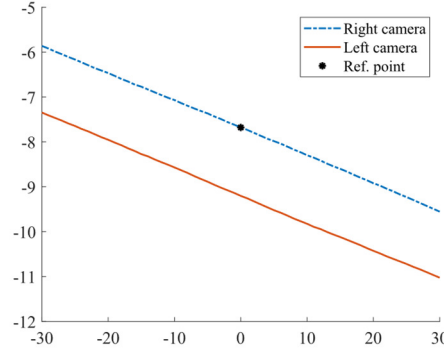


Figure 5: Phase matching between the right and left camera

In this scheme, for each point in the right camera, we look for a matching phase value in the left camera (within the search window) after adding the values from the stereo matching. **Figure 5** shows the matching process of a point in the right camera to the phase on the left camera after adding values from the stereo matching. Here we notice that the point (Ref. point) lies approximately within twenty pixels of the matching phase on the other camera. In this step, we find the point with the least difference from the matched point and correct the original map according to that. A 2D representation of the disparity map is shown in **Figure 6a**. A 3D representation of this refined map is shown in **Figure 6b**.

The refinement process results in filling the gaps in the disparity map and correct the mismatched points. The disparity map generated from this process results in a map with pixel-level resolution. To improve the resolution interpolation process was used. The reference image remains intact, while the second image is interpolated to provide more accurate points for matching. The results after using the interpolation can be seen in **Figure 6c**.

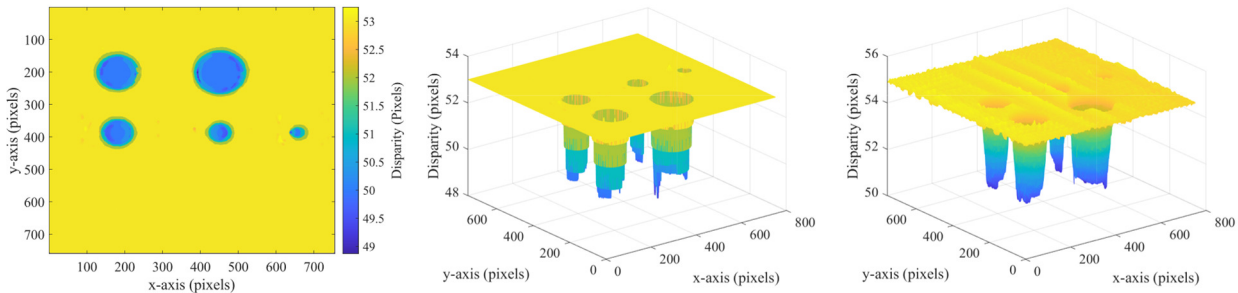


Figure 6: Reconstructed disparity map a) 2D representation of the refined surface, b) 3D representation of the refined surface, c) 3D representation of the interpolated refined surface

One of the problems that we faced during the experimental trials was the nonuniform phase distribution related to the barrel distortion from the projection lens (Wide projection is needed to cover large percentages of the two cameras field of view). This type of distortion results in a non-uniform phase shift across the acquired frame. Therefore, the reconstructed phase from the acquired image will not increase linearly.

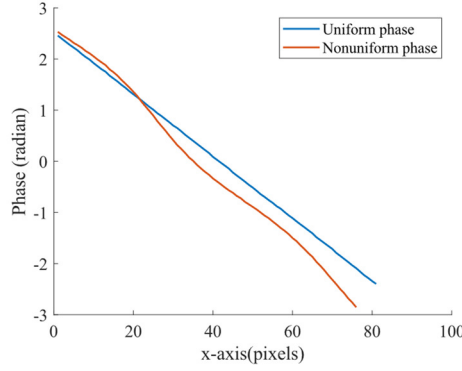


Figure 7: Effect of nonuniform phase distribution

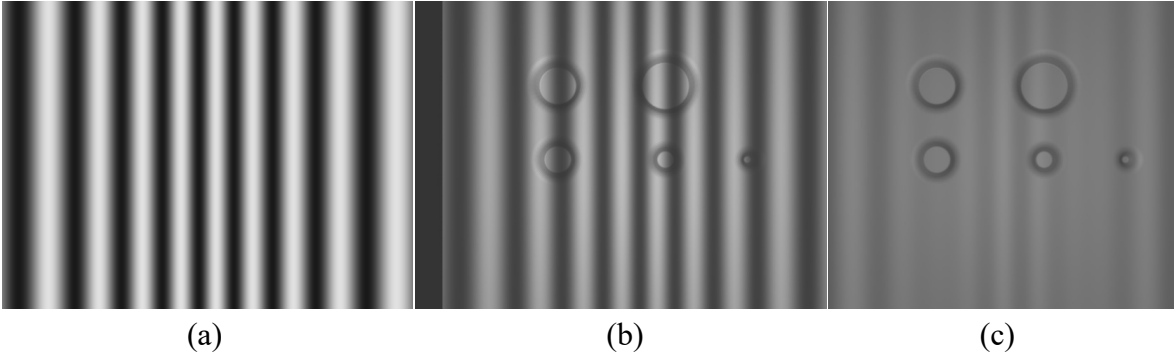


Figure 8: a) Nonlinear projected pattern, b) Sample image with nonlinear pattern projection Stereo pair with nonlinear projection, c) The average frame from the right camera

A comparison between the case of uniform and non-uniform phase projection is shown in **Figure 7**. We notice the nonlinear behavior of the calculated wrapped phase which will be reflected as a periodic pattern on the reconstructed surface. To replicate this effect we re-simulated the object with the nonlinear pattern shown in **Figure 8a**. A sample image from the simulated sequence is shown in **Figure 8b**. The DC component of the frames from the right camera is shown in **Figure 8c**. We notice that the DC image was reconstructed but not completely due to residues from the projected pattern. The pair of DC images are used to calculate the stereo disparity images and the results are shown in **Figure 9a**. The final reconstruction with the nonuniform distribution can be seen in **Figure 9b** and **c**. The 3D reconstruction shows that reconstruction was not affected by the nonuniform nature of the projected pattern. This is due to the fact that the stereo reconstruction doesn't depend on the relation between the pixels within a single frame. The stereo reconstruction depends on comparing the pixels from two different cameras.

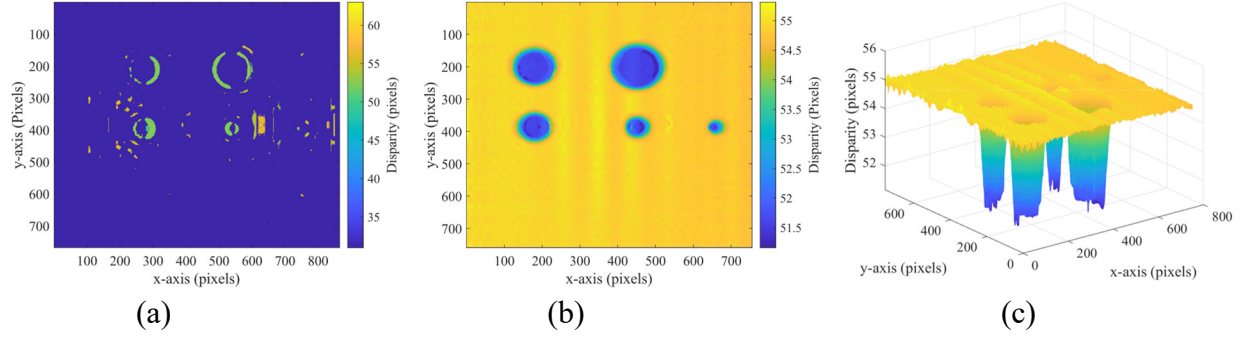


Figure 9: 3D reconstruction with nonlinear projection

1.2 Experimental results:

In this section, we apply the aforementioned enhancements to reconstruct the data from our structured light sensor shown in **Figure 10a**. Sample images from the structured light sensor are shown in **Figure 10b** and **c**. The images show strong vignetting in the projected frame with slight phase nonuniformity toward the edges of the frame. Averaging results to retrieve the DC images are shown in **Figure 11a** and **b**. These images can be used as an input for stereo matching and the machine learning algorithm. Direct reconstruction with stereo block matching is shown in **Figure 11c**. From the figure, we can clearly notice the existence of the defect but with a large number of outliers due to the small number of features on the flat surface. The refinement results of the disparity map are shown in **Figure 12a**. The refinement process of the experimental results creates a smaller number of outliers but introduces a nonlinear pattern to the scanning results. The pattern represents a periodic signal with a different phase shift for each row. By looking at the spectrum of a single row, we notice a peak at $\omega = 0.012$. **Figure 12c** shows the scanning results after applying a notch filter to suppress the periodic pattern. Our analysis of the results indicates that these artifacts are related to a small difference in the camera gamma values between the two cameras. To solve this issue, we are planning to calibrate the gamma of the cameras for future scanning tasks.

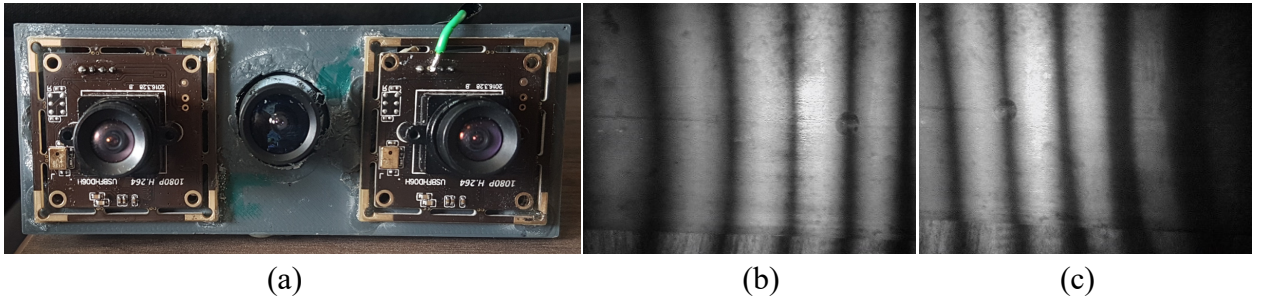


Figure 10: Experimental setup, a) Picture of the SL system, b) Sample image from the left camera, c) Sample image from the right camera

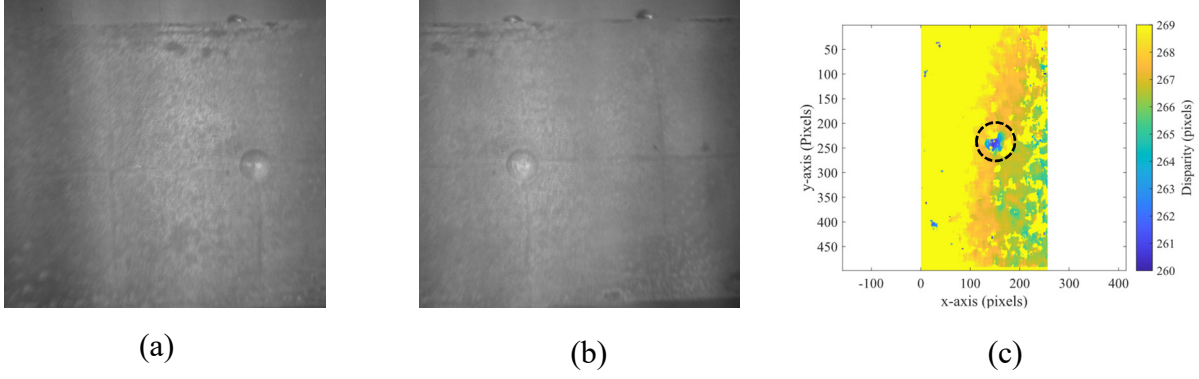


Figure 11: a) DC image from the left camera, b) DC image from the right camera, c) Disparity map from the direct stereo,

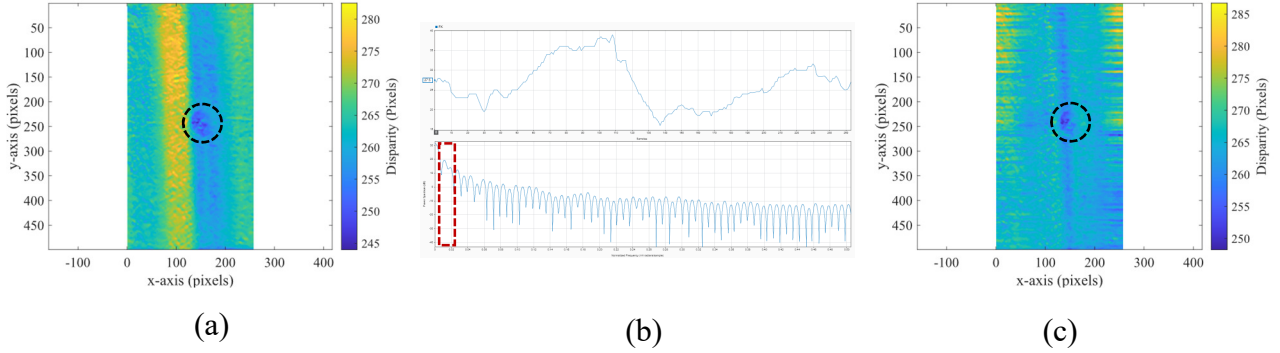


Figure 12: a) Refined disparity map by using phase information, b) Spatial representation and spectrum the 140th row, c) Filtered disparity map to remove artifacts.

1.3 Data collection:

One of the tasks that MSU is currently working on, is providing data to ASU to train the machine learning-based detection algorithm. The current plan is scan multiple pipes with different defect types and at different orientations to simulate real-world scenarios.

MSU NDEL laboratory has already developed two prototypes of the structured light modules from our previous work with GTI as shown in **Figure 13.a** and **Figure 13.b**. The first device is designed to scan pipes with a diameter of 5 to 8 inches while the second device is designed to scan pipes with a diameter of 2 to 4 inches. The data from these sensors are provided to offer initial training data for the training of the machine learning algorithm.

1. 6-inch polyvinyl chloride (PVC) pipe: an artificial defect with a rectangular shape was cut from the pipe wall and then the outer surface was covered with a white paper to enable the reconstruction of the shape of the defect. Internal and external images of the defect are shown in **Figure 14.a** and **Figure 14.b**. The pipe was



Figure 13: Assembled SL sensor, a) with the fisheye camera, b) with the omnidirectional camera

scanned by attaching the sensor to a linear scanner that moves in a parallel direction to the pipe main axis. Three hundred frames were collected at a rate of 15 frames/second (the robot moved at a speed of 0.8 inch/second). Each frame was reconstructed and then aligned and registered with the other consecutive frames. The final 3D reconstruction of the pipe section is shown in **Figure 14.d**. The 3D profile shows the 3D defect was detected and its profile was successfully recovered.

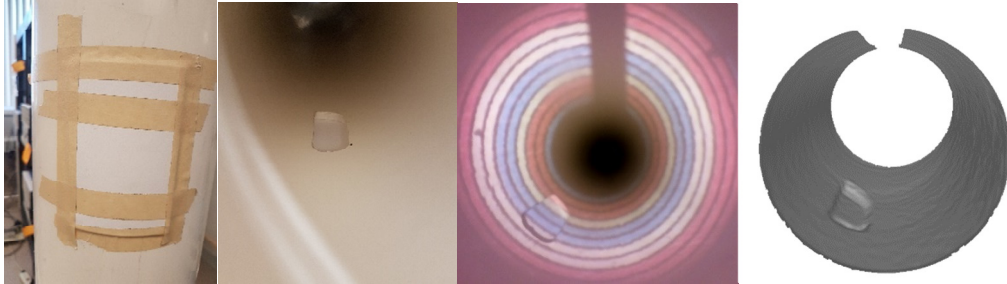


Figure 14: Scanning of PVC white pipe: a) External image of the defect, b) Internal image of the defect c) Sample image from the sensor showing the defect d) Final 3D rendering of the pipe surface

2. 6-inch MDPE: Another scan was performed to inspect a 6 inch HDPE pipe provided by the gas technology institute (GTI). The inspection system was used to detect and reconstruct artificially created volumetric defects on the pipe wall. **Figure 15.a** shows the scanning process and the reconstruction of an artificially created impingement defect. **Figure 15.b** shows the scanning process and the reconstruction of an artificially created wall loss defect in the circumferential direction. The 3D reconstruction of the entire pipe section is shown in **Figure 16.a**. To facilitate viewing the entire pipe section at once, the reconstructed profile is converted to the cylindrical domain to view the pipe as a single sheet as shown in **Figure 16.b**. In this view, we can easily identify both of the impingement and circumferential wall loss.

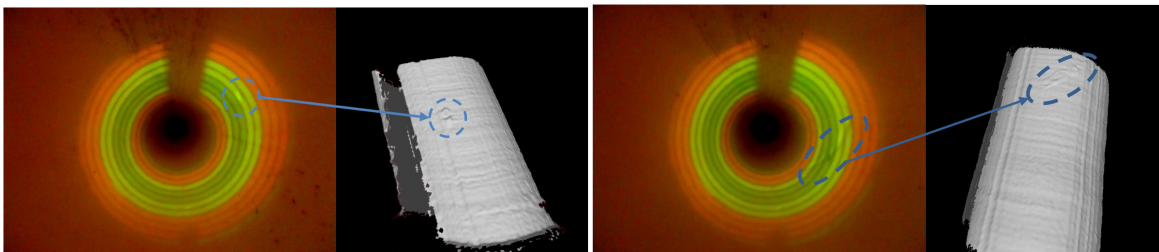


Figure 15: Single raw image 3D reconstructed pipe surface for a) Circular material loss, b) wall loss defect in the circumferential direction

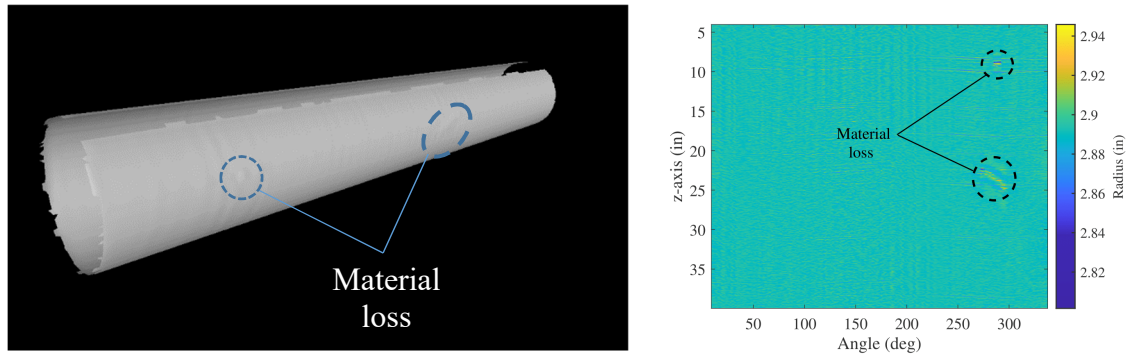


Figure 16: 35-inch long pipe section: a) 3D rendering, b) Cylindrical view

1.4 Summary:

In this quarter, MSU focused on developing and validating the stereo assisted phase-shifting framework with both simulations and experimental work. The algorithm was developed to use stereo cameras to circumvent the effect of the nonlinear projected phase. Stereo cameras are used to create an initial seed for the phase refinement. Geometrical constraints were also used to eliminate the need for phase unwrapping. The initial seed is refined by performing phase matching between the phase images from each camera. Subpixel resolution was achieved by performing interpolation and proved to be useful for enhancing the depth resolution. The developed framework was validated by using simulations with a POV ray simulator. The developed framework was later applied to our SL system for experimental validation and performance evaluation. The system performance was evaluated by scanning a defect with a depth of 2.1 mm. The system was able to successfully detect the defect but with some artifacts that are believed to be related to the cameras' gamma inconsistencies. MSU also started delivering data to ASU to train their machine learning model. Future work will focus on further system tuning and testing to enhance system performance and test the system limits. Another area under research is to exploit Fourier transform profilometry to enhance the image registration process and enable single-shot acquisition. This will be associated with continuing the work to collect the data for ASU from more samples.

MSU is still waiting for GTI to deliver the corroded test samples to have more realistic scanning data.

References:

- [1] W. Lohry, V. Chen, and S. Zhang, "Absolute three-dimensional shape measurement using coded fringe patterns without phase unwrapping or projector calibration," *Opt. Express*, vol. 22, no. 2, p. 1287, 2014.
- [2] G. Falcao, N. Hurtos, and J. Massich, "Plane-based calibration of a projector-camera system," 2008.
- [3] K. Konolige, "Small Vision Systems: Hardware and Implementation," *Robot. Res.*, pp. 203–212, 1998.

2 Task 3. Automated corrosion detection and uncertainty quantification

2.1 Overview:

The primary driver for Artificial Intelligence-based solutions for in-line inspection is the sheer size of piping infrastructure in refineries and cities. Full automation of the inspection procedure can reduce costs and drive higher reliability because automated inspection techniques are not susceptible to human error. The pipeline infrastructure is designed to last decades, and improper maintenance has significant safety and economic costs. The interactions between the fluids flowing through the pipe and the inner wall cause the degradation of the mechanical properties of the pipe over time, and early prognosis of damage can prevent catastrophic events. Vision-based tools for pipeline inspection is fast, cheap and does not always require a full-shutdown of the system for gas pipelines. The ability to obtain well-resolved visual analysis of pipelines on a regular basis can provide end-users critical information to plan their long-term maintenance strategy. Significant cost-savings can be achieved by improving upon visual inspection methods by implementing ever more efficient algorithms and intelligent post-processing methods. Fusing additional information effectively is a critical piece of designing algorithms that use multi-modal data. In this report, we improve upon the work done in the previous quarter and discuss how the progress made in this quarter can tie into further improvements and integration with structured-light data.

In the previous year, the focus was placed on the crack detection problem and efforts on corrosion detection began in the 4th quarter. We had procured test sets from real-world pipe inspections online, and the training set consisted of a texture dataset which contained both cracked and good concrete. We used a weakly-supervised region-proposal method to detect cracks in pipelines, where a supervised algorithm (random forest/CNN) was trained on the textures. In real world pipeline images, we used the Canny edge detector to find regions in the image that are most likely to have cracks. Those regions were passed into the trained supervised model and the detection pipeline was completed. This model did not require us to perform image-level annotations for training, but only required texture samples which can be easily procured. We also managed to obtain real-time performance with the Random Forest method. The weaknesses of this method were the following:

- The region proposal scheme based on the Canny edge detector was not class agnostic.
- The features in the random forest method were not enough to reduce the number of false detections.
- The CNN method, while having a much better classification performance, did not perform in real-time, due to the serial nature of the method architecture.
- The detections produced by this method had coordinate-level accuracy, but not pixel level accuracy.

The work in this quarter attempts to address some of these issues. We also started working on the corrosion detection problem. Weak localization cues for corrosion detection in the form of fractal dimension maps, a class agnostic saliency map obtained from the spectral domain were explored. We also implemented a pixel-level classification approach to corrosion detection, where we use Gaussian

Mixture Models (GMM) to obtain an initial segmentation of the images and discuss region merging schemes based on distance measures.

In this report, we use a class-agnostic, data-driven method for weak spatial localization cues for objects using a deep learning method. This method improves upon the feature-engineering based classical approaches. We use the data-driven localization cue as an information source for unsupervised segmentation using the GMM developed in the previous work. Standard benchmark datasets[1], [2] were used to train evaluate the segmentations, and the localization cues.

2.2 Localization Cues for Segmentation

2.2.1 Methodology

Deep learning methods for object recognition have seen heavy use for the past 8 years, owing to greater computational capacity and its performance in image recognition contests, where it outperformed all classical methods of object recognition given enough data. Neural Networks are given an objective function to train on. This objective function takes in some data and ground truth as input and calculates a loss. Minimizing this loss is key to finding well-tuned weights for the network. The objective for classification is far simpler than fully-supervised object detection. Algorithms such as YOLO[3], which perform fully-supervised object detection in a single pass through the network have composite loss functions that perform multiple tasks. It is interesting to note, however, that fully-supervised classification networks are a composition of operations over the image spatial domain. An input image is passed through several layers consisting of filters, and new representations are formed using these filters. After passing the image through multiple filters, a condensed representation of the image is obtained. This section exploits this representation to deliver coarse location cues for objects, given that the network is only trained for classification of images.

Neural networks can be considered universal function approximators. Given enough parameters and data, a neural network can, in theory, fit any continuous function in R^n . In practice, a single layer network overfits to the training data, and this led to the need for deeper networks. Compared to earlier CNNs which had less than 10 layers in total, deeper networks such as the VGG network architecture resulted in better performance. However, this improvement in performance reached its limits when the number of layers reached the order of 100, where the well-studied vanishing gradient problem comes into the picture. In essence, this is a performance reduction caused by information-loss, rather than overfitting. Backpropagation does not handle very deep networks well, and a solution was proposed by He et al. in [4]. Neural networks approximate functions based on the training data. Deeper networks with more layers can approximate a different class of functions as compared to shallow networks. A key problem in designing deep neural networks is that one does not know how many layers are needed to get to the right function class. Adding new layers changes the representations learned in an unpredictable fashion. Residual networks work by adding a so-called “skip connection” from one layer to another, deeper layer, as shown in **Figure 17**, resulting in a nested representation of functions.

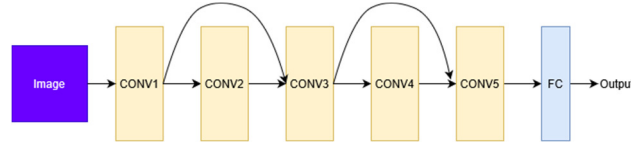


Figure 17: Conceptual ResNet architecture with 5 convolutional layers

Each residual block performs an “identity mapping”. The output of a layer L_k is fed into the input of layer L_{k+i} as an additive component to the output from layer L_{k+i-1} , as shown in **Figure 18**. Further discussion on identity mappings, residual networks and their mathematical properties are described in [4], [5].

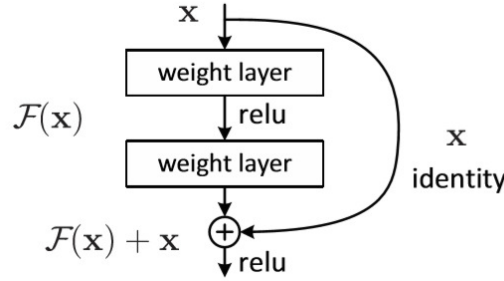


Figure 18: Residual identity mapping layer

The use of fully connected layers destroys the spatial dependencies that exist in the convolutional layers. Zhou et al [6] proposed a simple method to extract coarse object location information using Class Activation Maps (CAM) and Global Average Pooling (GAP) layers.

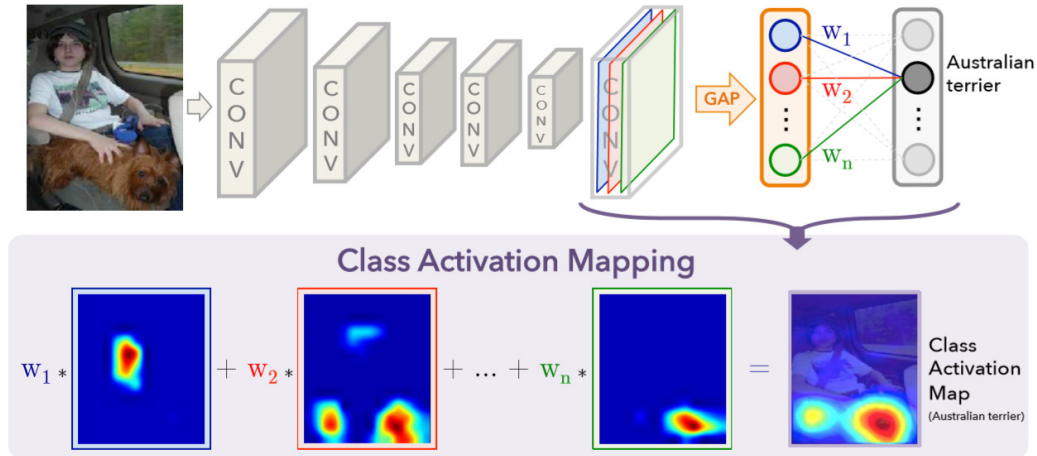


Figure 19 [6]: Schematic of the Class Activation Map extraction architecture

Figure 19 shows a schematic representation of the method. In our experiments, we use the ResNet, and the final fully connected layer is preceded by a global average pooling layer. Zhou et al in their paper describe the extraction of class activation maps as follows:

Let $f_k(x, y)$ represent the activation of unit k in the last convolutional layer at spatial location (x, y) . Then, for unit k , the result of global average pooling is:

$$F^k = \sum_{x,y} f_k(x,y)$$

The output from the global average pooling layer is passed to a softmax layer, where there are weight inputs for each class c . The weight is summed across all the unit outputs from the GAP layer:

$$S_c = \sum_k w_k^c F_k$$

The output of the softmax is a function with range $[0,1]$, acting as a probability:

$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}$$

The input to the softmax can also be written as:

$$S_c = \sum_{x,y} \sum_k w_k^c f_k(x,y)$$

Defining M_c as the class activation map for class c where each spatial element is given by:

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

$$S_c = \sum_{x,y} M_c(x,y)$$

M_c indicates the importance of each activation spatially. The size of this CAM representation, however, is compressed from the original image size. The architecture we employ for our task, for instance, gives a 7×7 map. This compressed representation needs to be upsampled to the original image size for a coarse localization, as shown in the **Figure 20**.

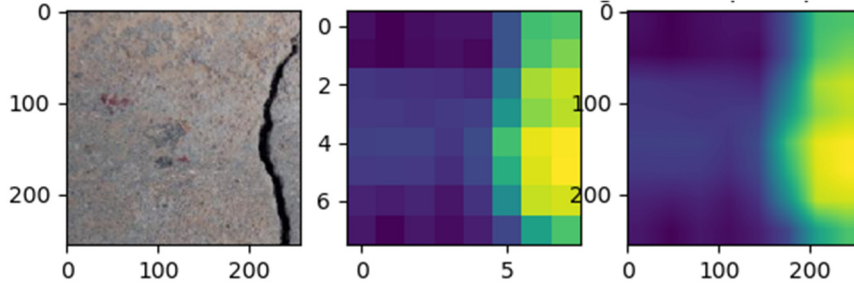


Figure 20: Class Activation Maps, original size (middle) and upsampled (right)

The end-goal for using the CAM was to detect weak location cues for objects in the scene. CAMs produce high activations for locations where it thinks an object of interest lies. However, one needs to modify the existing training methodology for generating cues for multiple objects of interest. The current formulation involves detecting one dominant objects in a scene that can possibly contain many objects. Consider a dataset with n classes. Let us say that an example scene has m , $m \leq n$ of those classes present, and each class c has c_k instances. The number of instances can be roughly estimated by the CAM if the objects of interest are well-separated and are distinct enough from the background. A case where this cannot happen are small-cracks dispersed throughout the image, which would be elaborated upon in the results. The objective then is to obtain m CAMs and combine the information obtained. Initially, the CAM outputs n CAMs, for all the classes, whether the class is indeed present in the image, as described

in the previous subsection. Out of the n outputs, only m are relevant for our task. The original method required that CAM is obtained through softmax operations. The softmax function provides us with a probability of the dominant class in the image. All the probabilities obtained through softmax add up to 1 across the classes in the image. Consider the sigmoid function:

$$f(x_i) = \frac{1}{1 + e^{-x_i}}$$

The output of the sigmoid function across all classes gives probabilities that are independent from one another, and therefore do not add to 1. In other words, the summing function in the denominator of the softmax constrains the sum across all classes to 1, where each class is treated separately by the sigmoid. The training process for the neural network is modified to accommodate the existence of multiple classes in the image by transforming the original label vector into a one-hot encoded matrix of dimension $Z^{N \times C}$, where N, C are the batch size and the number of classes respectively. We also incorporate a multi-label soft-margin loss function to

$$L(x, y) = -\frac{1}{C} \sum_i \frac{y[i]}{\log(1 + e^{-x[i]})} + (1 - y[i]) \log\left(\frac{e^{-x[i]}}{1 + e^{-x[i]}}\right)$$

$i \in \{0, \dots, |x|\}, y[i] \in \{0, 1\}$
 $|x| - \text{cardinality}$

Probability weighting: Since the sigmoid function is bounded between $[0, 1]$ and each output from the network is passed independently to the sigmoid, we can think of the sigmoid as giving us a class-wise probability. By multiplying each CAM element-wise with the probabilities, we can damp down all the non-existent maps to zero (or close to it).

$$PWCAM_i = p_i * CAM_i$$

Higher class activation values imply more significant pixels. Since insignificant maps are removed by the probability weighting, we can use the remaining original class activation maps to construct an overall map, with each class having a distinct label.

2.2.2 Results and discussion:

We use two benchmark datasets for our analysis, the first of which is a metal surface defect (MSD) dataset, and the second of which is a concrete crack (CC) defect dataset. The MSD dataset contains six defects: rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). RS and In are manufacturing defects, and the rest are material damage. **Figure 21** shows a sample of the dataset, which contains 1800 image samples in total, divided equally amongst the 6 defect classes.

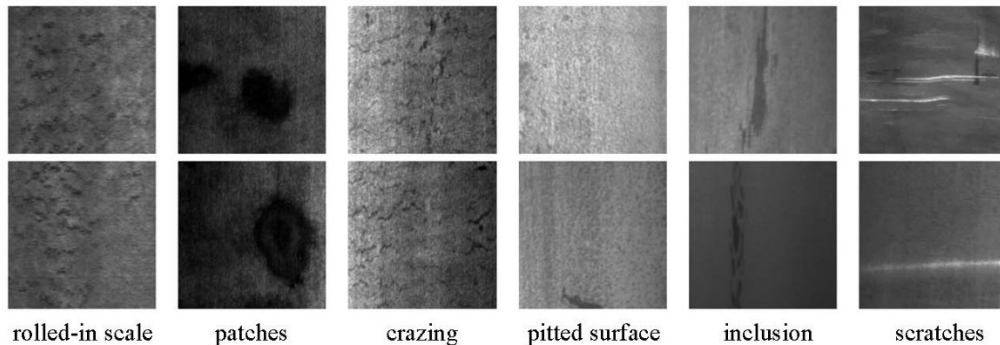


Figure 21: Representative images from the MSD dataset

The second dataset used is the binary crack dataset, which consists of cracked images and undamaged image samples. Each category has 1500 examples, with a few samples shown in **Figure 22**.



Figure 22: Representative images from the CC dataset

Assuming single defects, **Figure 23** shows the CAM, mask of the CAM using thresholding, and the original image. A key observation here is that the CAM only shows the activations that led to the CNN to decide on the class type for the image. This leads to the problem of missing small defects, cluttered defects across the scene and defects in cluttered scenes. This is observed clearly in the bottom left example (crazing) of **Figure 23**. The defect is covered by the entire image, by small cracks that are barely visible to the naked eye, and the CAM does not provide good cues. On the other hand, the CAM does a good job of isolating the location of both the corrosion patch and the scratch damage. The scratch damage is highly localized unlike the crazing defect, and this makes it easy for the CAM to detect it. Differences in the intensity of the scratch defect will not affect the CAM as the CNN learns how to detect the corresponding features using the training data.

CAM, Mask and Original Image

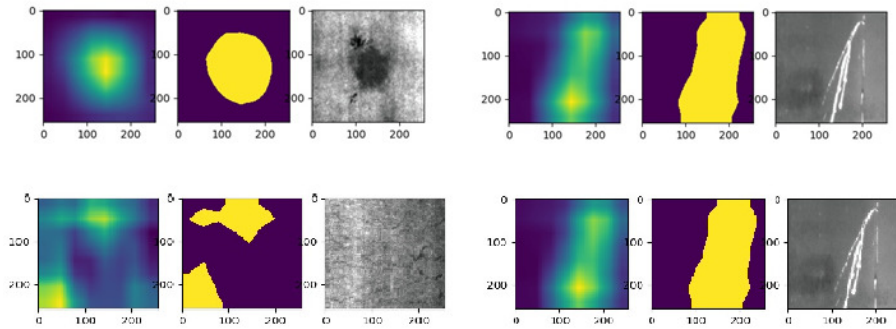


Figure 23: CAMs obtained from the MSD dataset

Figure 24 shows a comparison between the CAMs and probabilities produced by the softmax multi-class classification technique and the modified CAM with sigmoid and the multi-label soft margin loss. In **Figure 24 (a)**, we see that the probabilities add to 1, and the dominant class gets the highest probability. This dominant class is a function of both the number of occurrences and the training weights associated

with the softmax function. A higher weight during training for the corrosion patch class, would lend itself to producing higher probabilities for the patch class, even if the number of occurrences of this class is the same as another defect class. In **Figure 24** (a), on the right-hand side, we see that the scratch and the patch defects occur in equal numbers, but the patch defect is weighted higher due to a higher overall activation. **Figure 24** (b), in contrast shows the same examples when operated on by the sigmoid activation function. Each class is independent, and the probabilities do not add to 1. In the figure on the right-hand side, we see that the patch and scratch defects are represented with probabilities 0.64 and 0.61 respectively, and all the other classes have probabilities close to 0. This corresponds with the true class occurrences. However, the image on the left has 3 defects, namely, patch, scratch and inclusion. In this case, the sigmoid misses the inclusion defect. The errors made in the CAM can propagate through the algorithm, and it is one of the weaknesses that we are currently working to fix. The missing

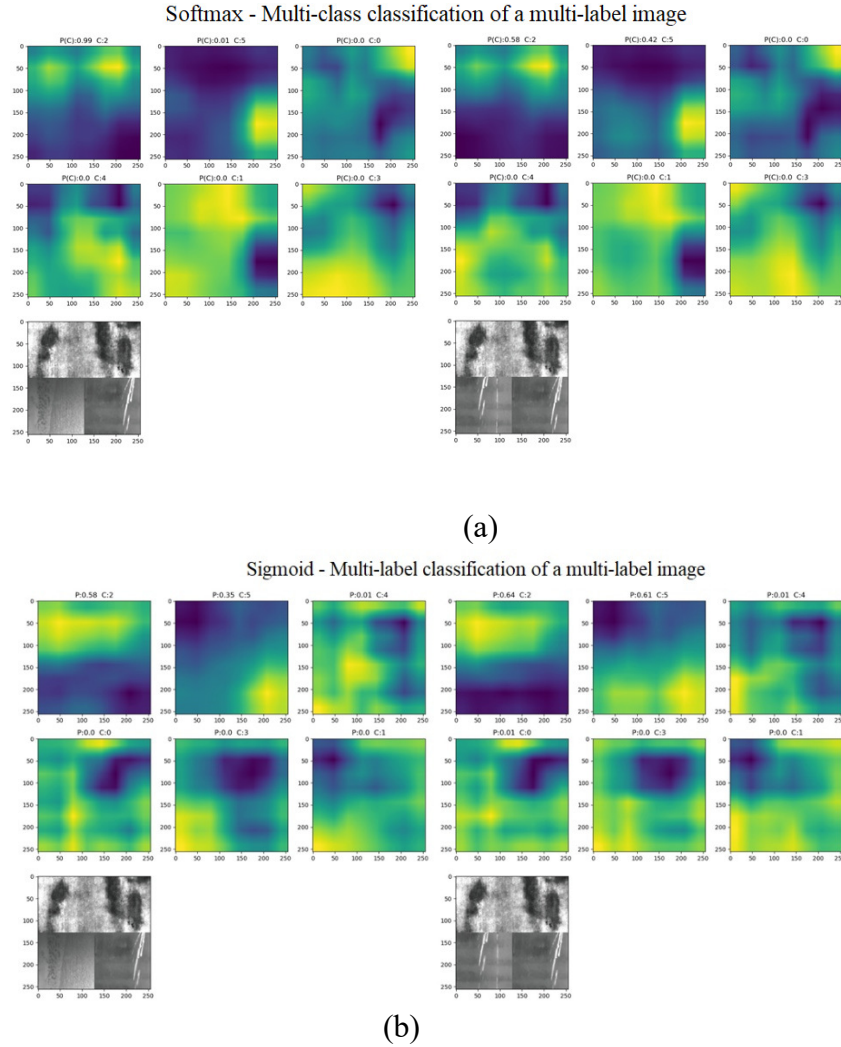


Figure 24: (a) CAMs obtained using the softmax operation and cross-entropy loss (b) CAMs obtained using the sigmoid operation and multi-label soft margin loss

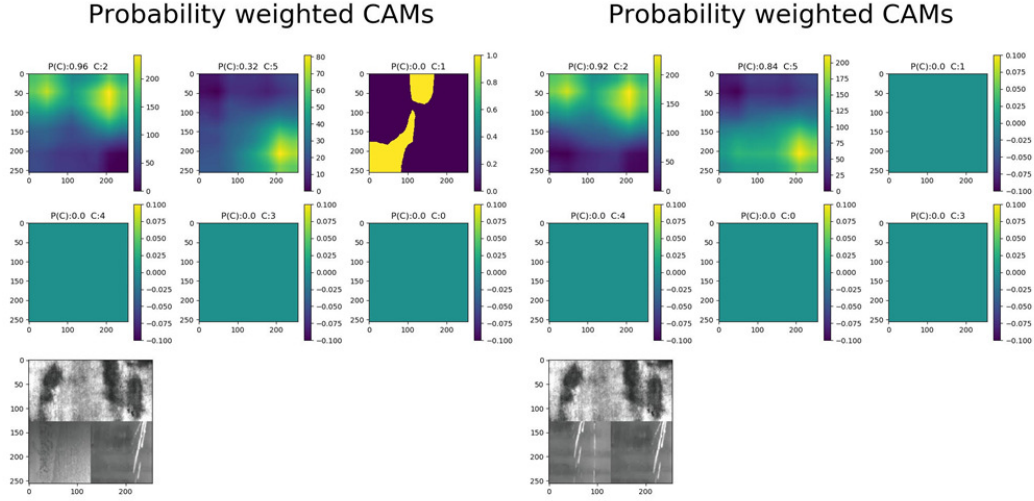


Figure 25: Probability Weighted CAMs for a multi-label image

Figure 25 shows the corresponding probability-weighted CAMs for the multilabel images. By multiplying out the sigmoid probabilities, any CAM with probability 0 automatically gets eliminated from consideration. The remaining low-probability CAMs are weighted by the respective maximum activation values after multiplication, and in our implementation, we threshold the maximum intensity to be above 50 for consideration. Doing so gives the best results for detecting existing classes, as the criterion for selecting the thresholding value is to not eliminate defects that might potentially be present. Increasing the threshold too much would end up removing classes that occur with less frequency or have much smaller activation than the dominant class, and too low a threshold might end up making the prediction too noisy. However, a higher threshold must be penalized more in this case.

2.3 Gaussian Mixture Model segmentation with Localization cues

2.3.1 Methodology:

We present first a short overview of the Gaussian Mixture Model (GMM) for segmenting images. Consider an image of size $N \times N$. Assume that each pixel is iid, resulting in a feature vector that is $N^2 \times 1$. The GMM defines a linear combination of K-Gaussians:

$$p(x) = \sum_{i=1}^K \pi_i N_i(x|\theta)$$

The mixtures form a convex combination, such that:

$$\begin{aligned} \pi_i &\geq 0 \\ \sum_{i=1}^K \pi_i &= 1 \end{aligned}$$

The parameters to be estimated are the K-mixture coefficients, along with the Gaussian parameters for each mixture, the mean μ and the covariance Σ . The dimensions of the means and the covariances depend on the number of features. In the case of a single-channel input, the mean and the covariances reduce to a scalar. The parameter vector ψ is estimated using an iterative Expectation Maximization algorithm.

$$\psi = \{\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K\}$$

Each pixel has associated with it a latent variable z , that is of dimension $K \times 1$ and takes a value 1 for the corresponding mixture index the pixel belongs to.

The initial estimates and cluster assignment are given by the k-means algorithm, and the prior vector for each pixel is:

$$\pi^0(z) = I_k$$

where I is the indicator variable for the k th cluster and z is the latent variable assigned to pixel x . The prior probabilities π_k are therefore initialized. After observing x , the posterior (responsibility) vector is calculated as:

$$\gamma(z_k) = p(z_k = 1|x) = \frac{\pi_k N(x|\theta_k)}{\sum_{j=1}^K \pi_j N(x|\theta_j)}$$

The above equation expresses the “responsibility” of the k th Gaussian to explain the occurrence of x . Since each pixel is assumed iid, the joint probability for the entire image consisting of N^2 pixels is the product of all individual pixel densities. Taking the log likelihood gives:

$$\ln p(X|\psi) = \sum_{j=1}^{N^2} \sum_{i=1}^K \pi_i N(x_j|\theta_i)$$

The E-step calculates this log-likelihood and the posterior update. The m-step maximizes the parameters by using a first order gradient equation. The update is given by:

$$\begin{aligned} \pi_k^{t+1} &= \frac{1}{K} \sum_{i=1}^K \gamma(z_{ik}) \\ \mu_k^{t+1} &= \frac{1}{K \pi_k^{t+1}} \sum_{i=1}^K x_i \gamma(z_{ik}) \\ \Sigma_k^{t+1} &= \frac{1}{K \pi_k^{t+1}} \sum_{i=1}^K (x_i - \mu_i^{t+1})^2 \gamma(z_{ik}) \end{aligned}$$

The update is performed until convergence of the log-likelihood to yield a local optimum. Initialization of the GMM can take place using either a random cluster centroid location, K-means or using custom centroids. The initial conditions determine the quality of the final solution, as the EM algorithm does not guarantee a global optimal solution, only locally optimal solutions. In this section we compute the initial clusters using the output from the probability weighted CAMs.

Consider a set of m CAMs obtained from the probability weighting process. The m CAMs are assumed here to contain the m class location cues. The GMM requires, for m clusters, a set of m means, variances and mixture weights. Currently, we equally weight the mixtures for each pixel. Therefore,

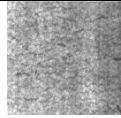
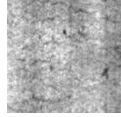
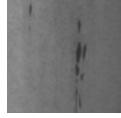
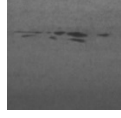
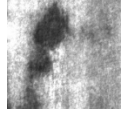
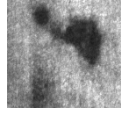
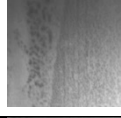
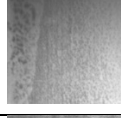
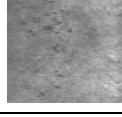
$$p(i, j) = \frac{1}{m}$$

$p(i, j)$ is the mixture probability scalar for each pixel in the image. The key idea is to fit a Gaussian to the obtained CAM data, to extract the corresponding means and variances. We extract a mask from the CAMs by using a thresholding operation, and consider the obtained indices as belonging to the class of

interest. Using these indices, we compute the means and variances of the pixel intensities in the original image.

2.3.2 *Results and Discussion:*

Table 1 shows a few demonstrative examples of the defects, and the corresponding initial conditions obtained through K-Means and the CAM gaussian fit for the GMM model. The results obtained **Figure 26** shows that for most images, the GMM converges to the same optimum for both the K-Means and the CAM-based initializations despite the differences in the initial values. This is because the differences in the cluster initialization is not big enough to cause a substantial shift in the local optimum of the EM algorithm.

Image	K-Means: Mean	K-Means: Variance	CAM: Means	CAM: Variance
	$\begin{pmatrix} 177.36 \\ 144.07 \end{pmatrix}$	$\begin{pmatrix} 484.57 \\ 378.37 \end{pmatrix}$	$\begin{pmatrix} 159.74 \\ 160.83 \end{pmatrix}$	$\begin{pmatrix} 493.32 \\ 764.51 \end{pmatrix}$
	$\begin{pmatrix} 189.47 \\ 157.20 \end{pmatrix}$	$\begin{pmatrix} 491.81 \\ 533.16 \end{pmatrix}$	$\begin{pmatrix} 181.43 \\ 172.06 \end{pmatrix}$	$\begin{pmatrix} 672.47 \\ 778.43 \end{pmatrix}$
	$\begin{pmatrix} 109.19 \\ 132.59 \end{pmatrix}$	$\begin{pmatrix} 114.33 \\ 40.04 \end{pmatrix}$	$\begin{pmatrix} 99.56 \\ 116.11 \end{pmatrix}$	$\begin{pmatrix} 219.31 \\ 164.85 \end{pmatrix}$
	$\begin{pmatrix} 91.66 \\ 108.83 \end{pmatrix}$	$\begin{pmatrix} 84.27 \\ 53.69 \end{pmatrix}$	$\begin{pmatrix} 97.85 \\ 107 \end{pmatrix}$	$\begin{pmatrix} 116.97 \\ 106.32 \end{pmatrix}$
	$\begin{pmatrix} 192.72 \\ 120.91 \end{pmatrix}$	$\begin{pmatrix} 1149.17 \\ 1850.85 \end{pmatrix}$	$\begin{pmatrix} 83.13 \\ 176.65 \end{pmatrix}$	$\begin{pmatrix} 930.29 \\ 1868.83 \end{pmatrix}$
	$\begin{pmatrix} 163.38 \\ 82.51 \end{pmatrix}$	$\begin{pmatrix} 813.70 \\ 999.43 \end{pmatrix}$	$\begin{pmatrix} 78.72 \\ 153.12 \end{pmatrix}$	$\begin{pmatrix} 1505.95 \\ 1436.56 \end{pmatrix}$
	$\begin{pmatrix} 150.07 \\ 101.07 \end{pmatrix}$	$\begin{pmatrix} 487.86 \\ 194.53 \end{pmatrix}$	$\begin{pmatrix} 183.62 \\ 123.10 \end{pmatrix}$	$\begin{pmatrix} 293.43 \\ 752.72 \end{pmatrix}$
	$\begin{pmatrix} 123.10 \\ 172.63 \end{pmatrix}$	$\begin{pmatrix} 452.34 \\ 674.01 \end{pmatrix}$	$\begin{pmatrix} 128.32 \\ 149.14 \end{pmatrix}$	$\begin{pmatrix} 694.94 \\ 1181.34 \end{pmatrix}$
	$\begin{pmatrix} 133.03 \\ 159.46 \end{pmatrix}$	$\begin{pmatrix} 146.31 \\ 236.60 \end{pmatrix}$	$\begin{pmatrix} 129.40 \\ 146.59 \end{pmatrix}$	$\begin{pmatrix} 156.96 \\ 350.93 \end{pmatrix}$

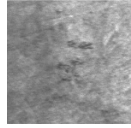
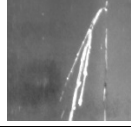

	$\begin{pmatrix} 131.53 \\ 167.917 \end{pmatrix}$	$\begin{pmatrix} 160.33 \\ 254.50 \end{pmatrix}$	$\begin{pmatrix} 151.24 \\ 145.82 \end{pmatrix}$	$\begin{pmatrix} 534.34 \\ 513.65 \end{pmatrix}$
	$\begin{pmatrix} 106.92 \\ 90.58 \end{pmatrix}$	$\begin{pmatrix} 154.81 \\ 82.35 \end{pmatrix}$	$\begin{pmatrix} 116.30 \\ 96.24 \end{pmatrix}$	$\begin{pmatrix} 169.12 \\ 154.25 \end{pmatrix}$
	$\begin{pmatrix} 97.86 \\ 161.31 \end{pmatrix}$	$\begin{pmatrix} 69.96 \\ 2604.37 \end{pmatrix}$	$\begin{pmatrix} 153.87 \\ 101.37 \end{pmatrix}$	$\begin{pmatrix} 2016.32 \\ 463.96 \end{pmatrix}$

Table 1: Comparison of initialization parameters on a few sample images using both K-Means and CAM, with the number of clusters inferred from the Probability Weighted CAM operations

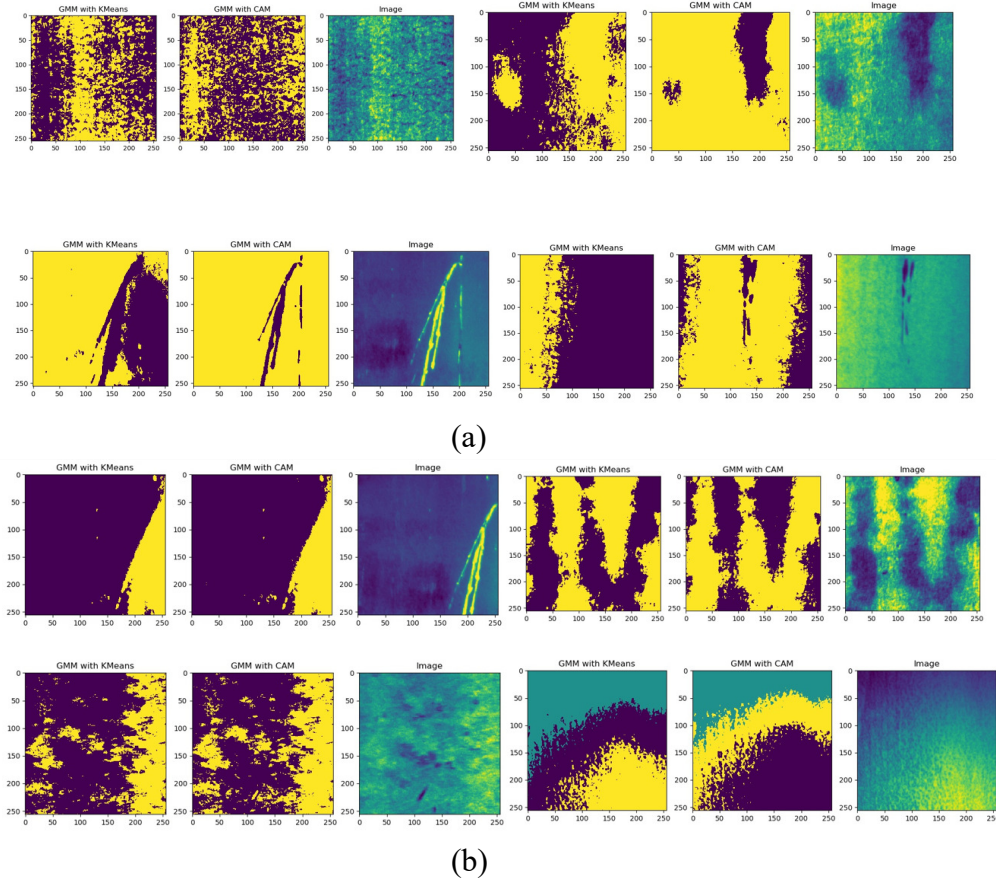
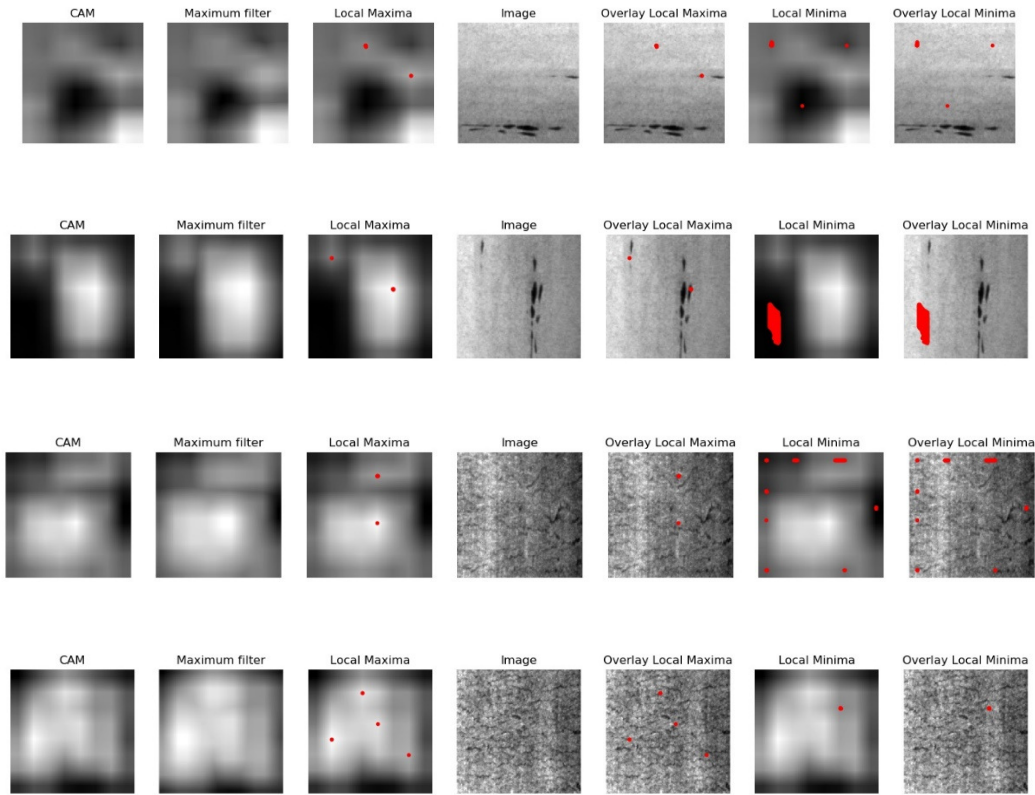


Figure 26: Demonstrative results for GMM segmentation with both K-Means initialization and CAM initialization (a) Cases where the CAM was better than K-Means (b) Cases where the CAM was worse or at best equal to the K-Means in terms of segmetation quality

The results in **Figure 26** lead to a conclusion that mere initialization with CAM values do not affect the GMM segmentation to increase detection performance. As explained in the preceding subsection, we extract the local extrema of the image to obtain interest points.

Region-growing segmentation methods and interactive segmentation methods make use of seed points for spatial cues. These seed points are often provided by the end-user or through a rule-based program. As we already have class activation maps, from which we can extract location cues, we compute image extrema using the activations obtained through the CAMs. A well-resolved CAM would have the highest activations in the regions where the object of interest is located. In the dataset under consideration, this would yield very well for 3 of the 6 defect types, namely patches, inclusion and scratches. Since the CAM did not yield good results for the other 3 smaller defects, we cannot rely upon the CAM to produce good seed points. The image extrema are computed using a local region-based maxima filtering method on the CAM. The foreground consists of the location cue for the object of interest, and the background is obtained by taking the inversion of the image using the bitwise-not operator. **Figure 27** shows the CAMs, and the corresponding local maxima and minima overlaid on the grayscale image. We see that a quality CAM provides accurate interest points for segmentation. These preliminary results show that we can take these initial points as “seeds” for segmentation. In the next quarter, we will expand upon this to explore whether seeds from CAMs would produce better segmentation results than initializing the GMM with clusters generated from CAMs.



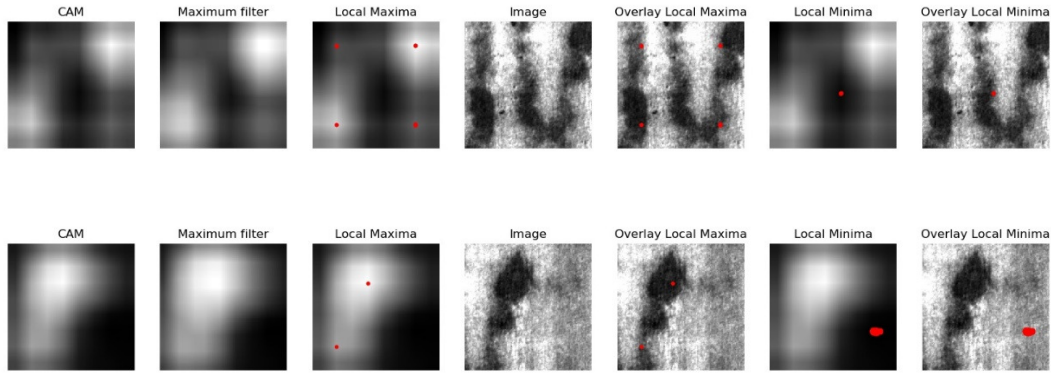


Figure 27: Seeding points obtained using the local extrema of the Class Activation Map

2.4 Conclusion and Future Work

In this report, we explored a method to obtain spatial cues for objects of interest in a scene through a data-driven deep learning procedure. We used the obtained class activation maps to calculate initializations for the Gaussian Mixture Model and found mixed results on its performance on the MSD dataset. We conclude that the k-means clustering initialization often converges to the same local optimum as the CAM-based initialization, as the differences in the initialization parameters are often not large enough to warrant a different optimum. The improvement in quality of CAM would lead to better initialization, and this would be a direction for research in the next quarter. We also propose a method to extract seeding points for segmentation using the CAMs, and even this could be aided by more accurate CAM representations. These seed points can be used by various segmentation techniques such as region-growing methods to cluster the scene. In the next quarter, we would also focus on using the point-cloud data obtained from MSU to integrate the color and 3-D information to produce improved detections. Image processing on color video data to remove the structured light pattern will also be an area of focus, to obtain quality color image data for segmentation.

References

- [1] K. Song and Y. Yan, “A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects,” *Applied Surface Science*, 2013, doi: 10.1016/j.apsusc.2013.09.002.
- [2] S. Dorafshan, R. J. Thomas, and M. Maguire, “SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks,” *Data in Brief*, 2018, doi: 10.1016/j.dib.2018.11.015.
- [3] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” 2018.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, doi: 10.1109/CVPR.2016.90.

- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, doi: 10.1007/978-3-319-46493-0_38.
- [6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, doi: 10.1109/CVPR.2016.319.